

Unit Testing for Backend

This is a file describing unit testing in backend including its purpose and corresponding functions in `backend/tests/`

Testing Google Generative API and integration with local class LLMModel (`test_genai_llmmodel.py`)

- User query history (`test_user_query_history_ability`): the API can retain information that previous provided by the user.
- API response time (`test_API_response_time`): the API can respond within 3 seconds.
- LLMModel instance extract the correct information from API response (`test_API_mock_response`)
- In case of API failure, LLMModel still outputs response (“..try again later”) without crashing the app (`test_API_send_message_error`)

Testing RAG Model (`test_qdrant_ragmodel.py`)

- If prompt-related information is not found: RAG model outputs `"I cannot find trusted source"` (`test_RAG_no_sourced_response`)
 - Input: “Tell me a joke”
 - Output: “I cannot find trusted source”
- If prompt-related information is found: RAG model outputs information extracting from `utils/` as source to respond the prompt (`test_RAG_with_source_response`)
 - Input: “What is blood pressure at hypertension stage 2?”
 - Output: “Hypertension Stage 2: systolic: 140 mm Hg or higher, diastolic: 90 mm Hg or higher” (*the format can be different but systolic and diastolic must keep the numbers*)

Testing fine-tuned model (`pre-trained model test.ipynb`)

- Test model answer before being fine-tuned.
- Test model answer after being fine-tuned.
- Criteria:
 - The model output is readable and understandable.
 - The output is related to training data.
 - Fine tuned model is accessible by the code